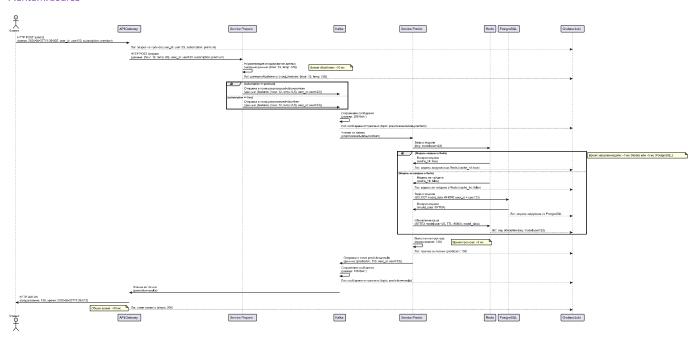
23.05.2025, 20:19 Wiki

Илья Дуванов

Процесс прогнозирования

Обновлено 22 мая 2025, 17:49

PlantumI source



Описание последовательности:

1. Клиент отправляет запрос на прогноз

- Клиент отправляет HTTP POST-запрос на эндпоинт /predict через API Gateway.
- Данные запроса включают параметры для прогноза (например, hour: 12, temp: 20), идентификатор пользователя (user_id: user123) и тип подписки (subscription: premium).
- Временная метка запроса: 2025-05-22T17:39:00Z.

2. API Gateway перенаправляет запрос в Service Prepare

- API Gateway маршрутизирует запрос в микросервис Service Prepare через HTTP POST-запрос на эндпоинт /prepare.
- Передаёт входные данные (hour: 12 , temp: 20), user_id И subscription .
- Логирование: Лог не создаётся на этом этапе, так как он уже зафиксирован на предыдущем шаге.

3. Service Prepare выполняет предобработку данных

- o Service Prepare нормализует и кодирует входные данные (например, temp: 20 преобразуется в нормализованное значение 0.5).
- ∘ Время обработки: ~10 мс.
- Логирование: Service Prepare отправляет лог в Grafana/Loki с параметрами: input_features: {hour: 12, temp: 0.5}.

4. Service Prepare отправляет данные в Kafka

- В зависимости от типа подписки выбирается топик Kafka:
 - Для premium-пользователя: preprocessed-data-premium.
 - Для free-пользователя: preprocessed-data-free .
- о Данные (features: {hour: 12, temp: 0.5}, user_id: user123) отправляются в топик preprocessed-data-premium.
- Kafka сохраняет сообщение (размер: 256 байт).
- о Логирование: Kafka отправляет лог в Grafana/Loki с параметром topic: preprocessed-data-premium.

5. Service Predict считывает данные из Kafka

- Service Predict асинхронно читает сообщение из топика preprocessed-data-premium.
- o Получает данные: features: {hour: 12, temp: 0.5}, user_id: user123.

6. Service Predict пытается загрузить модель из Redis

- o Service Predict запрашивает модель из Redis по ключу model:user123 (для premium-пользователя).
- Возможны два сценария:
 - **Модель найдена**: Redis возвращает модель (cache_hit: true). Время загрузки: ~1 мс.
 - Модель не найдена: Redis возвращает cache_hit: false.
- Логирование: Service Predict отправляет лог в Grafana/Loki: cache_hit: true или cache_hit: false.

7. Service Predict загружает модель из PostgreSQL (если не найдена в Redis)

- Если модель не найдена в Redis, Service Predict выполняет запрос к PostgreSQL: SELECT model_data WHERE user_id = user123.
- PostgreSQL возвращает сериализованную модель (model_data как вутеа).
- Время загрузки: ~5 мс.

23.05.2025, 20:19 Wiki

• Логирование: PostgreSQL отправляет лог в Grafana/Loki: "модель загружена из PostgreSQL".

8. Service Predict обновляет кэш в Redis

- После загрузки модели из PostgreSQL Service Predict сохраняет её в Redis с ключом model:user123 и TTL 24 часа (86400 секунд).
- Логирование: Redis отправляет лог в Grafana/Loki: key: model:user123.

9. Service Predict выполняет прогноз

- Service Predict использует загруженную модель для выполнения прогноза.
- o Результат: prediction: 150.
- Время прогноза: ~5 мс.
- Логирование: Service Predict отправляет лог в Grafana/Loki: prediction: 150.

10. Service Predict отправляет результат в Kafka

- Результат прогноза (prediction: 150, user_id: user123) Отправляется в топик prediction-results .
- Kafka сохраняет сообщение (размер: 128 байт).
- Логирование: Kafka отправляет лог в Grafana/Loki: topic: prediction-results.

11. API Gateway считывает результат из Kafka

- API Gateway асинхронно читает сообщение из топика prediction-results.
- o Получает данные: prediction: 150, user_id: user123.
- Объяснение: API Gateway ожидает результата через Kafka, что обеспечивает асинхронность и масштабируемость.

12. API Gateway возвращает результат клиенту

- API Gateway отправляет клиенту HTTP-ответ с кодом 200 ОК и результатом: prediction: 150.
- Временная метка ответа: 2025-05-22T17:39:01Z .
- ∘ Общее время обработки: ~50 мс.
- Логирование: API Gateway отправляет лог в Grafana/Loki: status: 200.